

UNITED STATES PATENT APPLICATION

for

**A FAST CODEBOOK SELECTION METHOD IN AUDIO ENCODING**

Applicant:

Jeongnam Youn  
Sundararam Dwarakanath  
Cheung Auyeung  
Takao Yamazaki

prepared by:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN  
12400 Wilshire Boulevard  
Los Angeles, CA 90026-1026  
(408) 720-8598

**EXPRESS MAIL CERTIFICATE OF MAILING**

"Express Mail" mailing label number EV336588710US

Date of Deposit September 29, 2003

I hereby certify that this paper or fee is being deposited with the United States Postal Service  
"Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above  
and is addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Michelle Begay

(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)

# **A FAST CODEBOOK SELECTION METHOD IN AUDIO ENCODING**

## **FIELD OF THE INVENTION**

**[0001]** The invention relates to audio encoding in general. More particularly, the invention relates to a selection of an optimal Huffman codebook for encoding of digital data.

## **COPYRIGHT NOTICE/PERMISSION**

**[0002]** A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever. The following notice applies to the software and data as described below and in the drawings hereto: Copyright © 2001, Sony Electronics, Inc., All Rights Reserved.

## **BACKGROUND OF THE INVENTION**

**[0003]** In recent years, efficient audio signal compression methods encode spectral data of quantized audio signals by using a Huffman encoding scheme. The standardized body, Motion Picture Experts Group (MPEG), discloses conventional data compression methods in their standards such as, for example, the MPEG-2 advanced audio coding (AAC) standard (see ISO/IEC 13818-7) and the MPEG-4 AAC standard (see ISO/IEC 14496-3). These standards are collectively referred to herein as the MPEG standard.

**[0004]** According to the MPEG standard, an input pulse code modulation (PCM) signal may be converted through a modified discrete cosine transform (MDCT) operation into 1024 elements of frequency spectral data that are quantized (converted into integers) using a non-linear quantization method. The 1024 elements of quantized frequency spectral data are then grouped into a number of scale factor bands (SFBs). Each SFB contains a multiple of 2 or 4 quantized spectral coefficients. The SFBs may also be grouped into sections.

**[0005]** Next, the quantized and grouped frequency spectral data is Huffman-encoded. The Huffman encoding operation is performed based on Huffman codebooks. Each section is encoded using a single Huffman codebook selected among eleven Huffman codebooks provided by the MPEG standard. Each Huffman codebook includes indexes of data items to be encoded, corresponding “codewords” (hexadecimal representations of encoded data items) and the code length of each codeword.

**[0006]** According to the MPEG standard, Huffman codebooks that can be selected for a section are limited by the largest absolute value (LAV) of the spectral data within the section. That is, the LAV of the spectral data has to be smaller than the LAV associated with a Huffman codebook that can be used for the section. Based on the MPEG standard, if the LAV of the spectral data within the section is less than 2, then the candidate Huffman codebooks are codebooks 0, 1 and 2. If the LAV of the spectral data within the section is less than 3, then candidate Huffman codebooks are codebooks 0 through 4. If the LAV of the spectral data within the section is less than 5, then candidate Huffman codebooks are codebooks 0 through 6. If the LAV of the spectral data within the section is less than 8, then candidate Huffman codebooks are codebooks 0 through 8.

If the LAV of the spectral data within the section is less than 13, then candidate Huffman codebooks are codebooks 0 through 10. Finally, if the LAV of the spectral data within the section is greater, or equal to 13, then candidate Huffman codebooks are codebooks 0 through 11.

[0007] Once the candidate Huffman codebooks are selected, the code length which would result when encoding data with each candidate Huffman codebook is determined and the results are compared. The candidate Huffman codebook that gives the smaller code length is selected and used for Huffman-encoding of the spectrum data in the section.

[0008] The above-described conventional method of selecting the optimal Huffman codebook for a section takes a large amount of computation because the operation of calculating the index and obtaining a corresponding code length must be performed separately for each of the candidate Huffman codebooks.

## **SUMMARY OF THE INVENTION**

[0009] The largest absolute value (LAV) is determined within a group of data. Based on the LAV, a difference table is identified that is to be used for selecting an optimal Huffman codebook for the group of data. The difference table is associated with two Huffman codebooks. Further, one or more indexes are calculated for the group of data using an expression associated with the two Huffman codebooks, and a size difference value is determined for the group of data using the calculated indexes and the

difference table. Based on the determined size difference value, the optimal Huffman codebook is selected from the above two codebooks.

### **BRIEF DESCRIPTION OF THE DRAWINGS**

[0010] The present invention will be understood more fully from the detailed description given below and from the accompanying drawings of various embodiments of the invention, which, however, should not be taken to limit the invention to the specific embodiments, but are for explanation and understanding only.

[0011] **Figure 1** is a block diagram of one embodiment of an encoding system.

[0012] **Figure 2** illustrates a table containing parameters of Huffman codebooks as provided by the MPEG standard.

[0013] **Figure 3** is a flow diagram of a codebook selection method, according to a prior art embodiment.

[0014] **Figure 4** is a flow diagram of one embodiment of a codebook selection process performed by a Huffman encoding module 114 of **Figure 1**.

[0015] **Figure 5** is a flow diagram of one embodiment of a codebook selection process used for MPEG-compliant audio encoding.

[0016] **Figure 6** illustrates the structure of a difference table according to one embodiment of the present invention.

[0017] **Figure 7** illustrates exemplary contents of difference tables 1 through 5, according to one embodiment of the present invention.

[0018] **Figure 8** is a block diagram of a computer environment suitable for practicing embodiments of the present invention.

## DETAILED DESCRIPTION OF THE INVENTION

[0019] In the following detailed description of embodiments of the invention, reference is made to the accompanying drawings in which like references indicate similar elements, and in which is shown, by way of illustration, specific embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be utilized and that logical, mechanical, electrical, functional and other changes may be made without departing from the scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims.

[0020] Beginning with an overview of the operation of the invention, **Figure 1** illustrates one embodiment of an encoding system 100. The encoding system 100 is in compliance with MPEG audio coding standards (e.g., the MPEG-2 AAC standard, the MPEG-4 AAC standard, etc.) that are collectively referred to herein as the MPEG standard. The encoding system 100 includes a filterbank module 102, coding tools 104, a psychoacoustic modeler 106, a quantization module 110, and a Huffman encoding module 114.

[0021] The filterbank module 102 receives a pulse code modulation (PCM) signal, modulates it using a window function, and then performs a modified discrete cosine transform operation (MDCT). The window function modulates the signal using two types of operation, one being a long window type in which a signal to be analyzed is expanded in time for improved frequency resolution, the other being a short window type

in which a signal to be analyzed is shortened in time for improved time resolution. The long window type is used in the case where there exists only a stationary signal, and the short window type is used when there is a rapid signal change. By using these two types of operation according to the characteristics of a signal to be analyzed, it is possible to prevent the generation of unpleasant noise called a pre-echo, which would otherwise result from an insufficient time resolution. The MDCT operation is performed to convert the time-domain signal into a number of samples of frequency spectral data.

[0022] The coding tools 104 include a set of optional tools for spectral processing. For example, the coding tools may include a temporal noise shaping (TNS) tool and a prediction tool. The TNS tool may be used to control the temporal shape of the noise within each window of the transform and to solve the pre-echo problem. The prediction tool may be used to remove the correlation between the samples.

[0023] The psychoacoustic modeler 106 analyzes the samples to determine an auditory masking curve. The auditory masking curve indicates the maximum amount of noise that can be injected into each respective sample without becoming audible. What is audible in this respect is based on psychoacoustic models of human hearing. The auditory masking curve serves as an estimate of a desired noise spectrum.

[0024] The quantization module 110 is responsible for quantizing the samples using optimal scale factors (also known as quantizers). The optimal scale factors are determined using rate control 108 and distortion control 112. The rate control 108 is performed in accordance with the required bit-rate, and the distortion control 112 is performed in accordance with the desired noise spectrum as defined by the masking curve provided by the psychoacoustic modeler 106. The resulting quantized spectral

coefficients are grouped into scalefactor bands (SFBs). Each SFB includes coefficients that resulted from the use of the same quantizer. In one embodiment, each SFB contains a multiple of 4 quantized spectral coefficients.

**[0025]** The Huffman encoding module 114 is responsible for selecting an optimal Huffman codebook for each group of quantized spectral coefficients and performing the Huffman-encoding operation using the optimal Huffman codebook. The resulting variable length code (VLC) and data identifying the codebook used in the encoding (and some other information when necessary) are subsequently assembled into a bit stream.

**[0026]** In one embodiment, eleven Huffman codebooks are provided by the MPEG standard for selection. Each Huffman codebook includes indexes of data items to be encoded, corresponding “codewords” (hexadecimal representations of encoded data items) and the code length of each codeword. Candidate Huffman codebooks that can be selected for a group of spectral data are limited by the largest absolute value (LAV) of the spectral data within the group. That is, the LAV of the spectral data has to be smaller than the LAV associated with a candidate Huffman codebook. As a result, the number of candidate Huffman codebooks that are to be used for a selection of an optimal codebook for a group of data varies, depending on the LAV within the group of data, from two candidate codebooks for the LAV less than 2 to eleven candidate codebooks for the LAV greater than, or equal to, 13.

**[0027]** As will be discussed in more detail below, the Huffman encoding module 114 limits the number of candidate Huffman codebooks for a group of data to 2 by creating a set of difference tables, pre-computing their content and storing them in



memory for use during Huffman encoding operations. Each difference table is associated with two distinct Huffman codebooks. In one embodiment, each entry in a difference table contains a length difference value that is a difference between corresponding code length entries from the two Huffman codebooks associated with the difference table.

[0028] A conventional codebook selection method will now be discussed in more detail to illustrate the advantages of the operation of the Huffman encoding module 114.

[0029] **Figure 2** illustrates a table 200 containing parameters of Huffman codebooks as provided by the MPEG standard.

[0030] Referring to **Figure 2**, the table 200 provides codebook number information 202, sign information 204, codebook dimension information 206, codebook LAV information 208, and codebook table information 210. The codebook dimension information 206 specifies whether a Huffman codebook encodes 2- or 4-tuples of quantized spectral coefficients. The codebook sign information 204 indicates whether a Huffman codebook can represent signed (e.g.,  $\text{unsigned\_cb}[i] = 0$ ) or unsigned (e.g.,  $\text{unsigned\_cb}[i] = 1$ ) spectral data. If an unsigned codebook is selected, a codeword extracted from the codebook for each non-zero coefficient in the group of data is provided with sign bits that immediately follow the codeword.

[0031] A group of data may be an SFB or a section. A section is formed by one or more SFBs that are represented by a single Huffman codebook.

[0032] **Figure 3** is a flow diagram of a codebook selection method 300 according to a prior art embodiment. Method begins 300 with finding all candidate codebooks for a group of data based on the LAV within the group (processing block 302).

As illustrated in the table 200, if the LAV within the group is less than 2, then the candidate Huffman codebooks are codebooks 0, 1 and 2. If the LAV within the group is less than 3, then candidate Huffman codebooks are codebooks 0 through 4. If the LAV within the group is less than 5, then candidate Huffman codebooks are codebooks 0 through 6. If the LAV within the group is less than 8, then candidate Huffman codebooks are codebooks 0 through 8. If the LAV within the group is less than 13, then candidate Huffman codebooks are codebooks 0 through 10. Finally, if the LAV within the group is greater than, or equal to, 13, then candidate Huffman codebooks are codebooks 0 through 11.

**[0033]** Next, a first codebook is selected from the candidate codebooks for processing (processing block 304).

**[0034]** At processing block 306, the dimension of the selected codebook is checked in the table 200 to determine whether this codebook encodes 2- or 4-tuples of coefficients and, accordingly, the first 2- or 4-tuples of coefficients are identified within the group of data.

**[0035]** At processing block 308, a codebook index is computed for these 2- or 4-tuples of coefficients using an expression associated with the selected codebook. Based on the MPEG standard, the expressions for computing the codebook index are as follows:

(a) for codebooks 1 and 2

$$\text{codebook\_index} = 27 * C[i] + 9 * C[i+1] + 3 * C[i+2] + C[i+3] + 46,$$

(b) for codebooks 3 and 4

$$\text{codebook\_index} = 27 * \text{abs}(C[i]) + 9 * \text{abs}(C[i+1]) + 3 * \text{abs}(C[i+2]) + \text{abs}(C[i+3]),$$

(c) for codebooks 5 and 6

$$\text{codebook\_index} = 9 * C[i] + C[i+1] + 40,$$

(b) for codebooks 7 and 8

$$\text{codebook\_index} = 8 * \text{abs}(C[i]) + \text{abs}(C[i+1]),$$

(b) for codebooks 9 and 10

$$\text{codebook\_index} = 13 * \text{abs}(C[i]) + \text{abs}(C[i+1]),$$

wherein  $C[i]$ ,  $C[i+1]$ ,  $C[i+2]$  and  $C[i+3]$  are values of quantized spectral coefficients within the group of data.

**[0036]** If the LAV within the group of data is greater than, or equal to, 12, the expressions for computing the codebook index are as follows:

(a) if  $\text{abs}(C[i]) \geq 16$  and  $\text{abs}(C[i+1]) \geq 16$

$$\text{codebook\_index} = 272 + \text{abs}(C[i]),$$

(b) if  $\text{abs}(C[i]) \geq 16$  and  $\text{abs}(C[i+1]) < 16$

$$\text{codebook\_index} = 17 * \text{abs}(C[i]) + 16,$$

(c) if  $\text{abs}(C[i]) < 16$  and  $\text{abs}(C[i+1]) \geq 16$

$$\text{codebook\_index} = 17 * \text{abs}(C[i]) + \text{abs}(C[i+1]).$$

**[0037]** Once the index is computed using one of the above expressions, the selected codebook is accessed using the index to obtain the length of a corresponding codeword (processing block 310). This length is then added to a total length value (which is initially equal to 0) accumulated for this codebook (processing block 312).

**[0038]** Next, a determination is made as to whether there are more unprocessed 2- or 4-tuples of coefficients in the group (decision box 314). If so, the next 2- or 4-tuples of coefficients are identified (processing block 316), and method 300

returns to processing block 308. Once all the coefficients in the group are covered, method 300 selects a next unchecked codebook from the candidate codebooks (processing block 320) and repeats processing of blocks 306 through 314. The processing of these blocks is repeated until all candidate codebooks are checked (decision box 318). Then, the total length values accumulated for the candidate codebooks are compared (processing block 322) and the codebook that produces the shortest codeword is chosen as an optimal codebook for this group of spectral data (processing block 324).

**[0039]** Thus, the above-described conventional method repeats processing blocks 306 through 318 for each candidate codebook table, which may result in up to 11 repetitions.

**[0040]** The Huffman encoding module 114 of **Figure 1** eliminates these repetitions by creating a set of difference tables, pre-computing their content and storing them in memory for use during Huffman encoding operations. As discussed above, each difference table is associated with two distinct Huffman codebooks. For example, difference table 1 may be associated with codebooks 1 and 2, difference table 2 may be associated with codebooks 3 and 4, difference table 3 may be associated with codebooks 5 and 6, difference table 4 may be associated with codebooks 7 and 8, and difference table 5 may be associated with codebooks 9 and 10. In one embodiment, each entry in a difference table contains a length difference value that is a difference between corresponding code length entries from the two Huffman codebooks associated with the difference table.

**[0041]** **Figures 4 and 5** are flow diagrams of two embodiments of a codebook selection process that may be performed by a Huffman encoding module 114 of **Figure 1**.

The process may be performed by processing logic that may comprise hardware (e.g., circuitry, dedicated logic, etc.), software (such as run on a general purpose computer system or a dedicated machine), or a combination of both. For software-implemented processes, the description of a flow diagram enables one skilled in the art to develop such programs including instructions to carry out the processes on suitably configured computers (the processor of the computer executing the instructions from computer-readable media, including memory). The computer-executable instructions may be written in a computer programming language or may be embodied in firmware logic. If written in a programming language conforming to a recognized standard, such instructions can be executed on a variety of hardware platforms and for interface to a variety of operating systems. In addition, the embodiments of the present invention are not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings described herein. Furthermore, it is common in the art to speak of software, in one form or another (e.g., program, procedure, process, application, module, logic...), as taking an action or causing a result. Such expressions are merely a shorthand way of saying that execution of the software by a computer causes the processor of the computer to perform an action or produce a result. It will be appreciated that more or fewer operations may be incorporated into the processes illustrated in **Figures 4 and 5** without departing from the scope of the invention and that no particular order is implied by the arrangement of blocks shown and described herein.

[0042] Referring to **Figure 4**, processing logic begins with identifying a group of data (e.g., a SFB, a section, etc.) to be processed (processing block 402). Next,

processing logic determines the LAV within the group of data (processing block 404) and identifies a difference table to be used for this group of data based on the LAV (processing block 406). The difference table is associated with two specific Huffman codebooks that are to be considered in the selection of an optimal codebook for this group of data. In one embodiment, processing logic identifies the difference table based on the predefined correspondence between the LAV and the difference table. In one embodiment, the predefined correspondence is determined via the associated candidate Huffman codebooks that are, in turn, linked to the LAV based on statistical data indicating that these two candidate codebooks have the highest likelihood of being an optimal codebook for this LAV. In another embodiment, processing logic identifies the difference table by dynamically determining the correspondence between the LAV and the difference table during each codebook selection operation. The determination may be based on statistical data collected during prior codebook selection operations.

[0043] At processing block 408, processing logic calculates an index for first  $n$ -tuples of data values within the group of data using an expression associated with the two candidate Huffman codebooks. The number  $n$  depends on the dimension of the candidate Huffman codebooks as defined, for example, by table 200 of **Figure 2**. In one embodiment, the expression is constructed to eliminate multiplication operations, thus accelerating the index computation process. In one embodiment, the expressions associated with different pairs of Huffman codebooks were developed based on the expressions provided by the MPEG standard and the operations involved in computing and structuring the difference tables.

[0044] Next, processing logic accesses the difference table using the calculated index to retrieve a corresponding size difference value (processing block 410) and adds this value to a total difference value (processing block 412).

[0045] At decision box 414, processing logic determines whether the group of data includes more unprocessed  $n$ -tuples of data values. If so, processing logic identifies next  $n$ -tuples within the group of data (processing block 418) and repeats processing blocks 410 through 414. This processing is repeated until all data values in the group are processed.

[0046] Further, processing logic selects an optimal Huffman codebook from the two candidate Huffman codebooks based on the total difference value (processing block 416). In one embodiment, processing logic selects a first of the two candidate Huffman codebooks if the total difference value is greater than zero. Otherwise, processing logic selects a second of the two candidate Huffman codebooks.

[0047] **Figure 5** is a flow diagram of one embodiment of a codebook selection process 500 used for MPEG-compliant audio encoding. Based on the MPEG standard, 11 Huffman codebooks are provided for selecting an optimal codebook for each section.

[0048] Referring to **Figure 5**, processing logic begins with identifying a section to be processed (processing block 502). A section may include one or more SFBs, with each SFB having a multiple of 4-tuples of quantized spectral coefficients.

[0049] Next, processing logic determines the LAV within the section (processing block 404).

[0050] At decision box 506, processing logic determines whether the LAV is equal to zero. If so, processing logic selects codebook 0 to Huffman encode the section

(processing block 508). If not, processing logic determines whether the LAV is less than 2 (decision box 510). If so, processing logic calculates the index for each  $n$ -tuples of quantized spectral coefficients using expression 1 associated with Huffman codebooks 1 and 2 (processing block 512). In one embodiment, expression 1 is defined as

$$\text{codebook\_index} = C[i] \ll 6 + C[i+1] \ll 4 + C[i+2] \ll 2 + C[i+3] + 85.$$

**[0051]** Next, processing logic accesses difference table 1 using the index for each  $n$ -tuples of quantized spectral coefficients to obtain a corresponding bit difference value (processing block 514). Each entry in the difference table 1 (DIFF\_TAB\_1) is a difference in length of the corresponding codewords from the codebooks 1 and 2.

**[0052]** At processing block 516, processing logic determines a total bit difference value. If the total bit difference value is greater than zero (decision box 518), processing logic selects codebook 2 (processing block 520). Otherwise, processing logic selects codebook 1 (processing block 522).

**[0053]** If processing logic determines that LAV is less than 3 (decision box 524), processing logic calculates the index for each  $n$ -tuples of quantized spectral coefficients using expression 2 associated with Huffman codebooks 3 and 4 (processing block 526). In one embodiment, expression 2 is defined as

$$\text{codebook\_index} = \text{abs}(C[i]) \ll 6 + \text{abs}(C[i+1]) \ll 4 + \text{abs}(C[i+2]) \ll 2 + \text{abs}(C[i+3]).$$

**[0054]** Next, processing logic accesses difference table 2 using the index for each  $n$ -tuples of quantized spectral coefficients to obtain a corresponding bit difference value (processing block 528). Each entry in the difference table 2 (DIFF\_TAB\_2) is a difference in length of the corresponding codewords from the codebooks 3 and 4.



[0055] At processing block 530, processing logic determines a total bit difference value. If the total bit difference value is greater than zero (decision box 532), processing logic selects codebook 4 (processing block 534). Otherwise, processing logic selects codebook 3 (processing block 536).

[0056] If processing logic determines that the LAV is less than 5 (decision box 538), processing logic calculates the index for each  $n$ -tuples of quantized spectral coefficients using expression 3 associated with Huffman codebooks 5 and 6 (processing block 540). In one embodiment, expression 3 is defined as

$$\text{codebook\_index} = C[i] \ll 4 + C[i+1] + 68.$$

[0057] Next, processing logic accesses difference table 3 using the index for each  $n$ -tuples of quantized spectral coefficients to obtain a corresponding bit difference value (processing block 542). Each entry in the difference table 3 (DIFF\_TAB\_3) is a difference in length of the corresponding codewords from the codebooks 5 and 6.

[0058] At processing block 544, processing logic determines a total bit difference value. If the total bit difference value is greater than zero (decision box 546), processing logic selects codebook 6 (processing block 548). Otherwise, processing logic selects codebook 5 (processing block 550).

[0059] If processing logic determines that the LAV is less than 8 (decision box 552), processing logic calculates the index for each  $n$ -tuples of quantized spectral coefficients using expression 4 associated with Huffman codebooks 7 and 8 (processing block 554). In one embodiment, expression 4 is defined as

$$\text{codebook\_index} = \text{abs}(C[i]) \ll 3 + \text{abs}(C[i+1]).$$

[0060] Next, processing logic accesses difference table 4 using the index for each  $n$ -tuples of quantized spectral coefficients to obtain a corresponding bit difference value (processing block 556). Each entry in the difference table 4 (DIFF\_TAB\_4) is a difference in length of the corresponding codewords from the codebooks 7 and 8.

[0061] At processing block 558, processing logic determines a total bit difference value. If the total bit difference value is greater than zero (decision box 560), processing logic selects codebook 8 (processing block 562). Otherwise, processing logic selects codebook 7 (processing block 564).

[0062] If processing logic determines that the LAV is less than 13 (decision box 566), processing logic calculates the index for each  $n$ -tuples of quantized spectral coefficients using expression 5 associated with Huffman codebooks 9 and 10 (processing block 568). In one embodiment, expression 5 is defined as

$$\text{codebook\_index} = \text{abs}(C[i]) \ll 4 + \text{abs}(C[i+1]).$$

[0063] Next, processing logic accesses difference table 5 using the index for each  $n$ -tuples of quantized spectral coefficients to obtain a corresponding bit difference value (processing block 570). Each entry in the difference table 5 (DIFF\_TAB\_5) is a difference in length of the corresponding codewords from the codebooks 9 and 10.

[0064] At processing block 572, processing logic determines a total bit difference value. If the total bit difference value is greater than zero (decision box 574), processing logic selects codebook 10 (processing block 576). Otherwise, processing logic selects codebook 9 (processing block 578).

[0065] If processing logic determines at decision box 566 that the LAV is not less than 13, then processing logic selects codebook 11 to Huffman encode the section (processing block 580).

[0066] In one embodiment, each difference table stores bit difference values as a list of entries. As a result, each computed index represents an order number and points to an entry associated with this order number within the list. **Figure 6** illustrates the structure of a difference table according to one embodiment of the present invention.

[0067] **Figure 7** illustrates exemplary contents of difference tables 1 through 5, according to one embodiment of the present invention.

[0068] The following description of **Figure 8** is intended to provide an overview of computer hardware and other operating components suitable for implementing the invention, but is not intended to limit the applicable environments. **Figure 8** illustrates one embodiment of a computer system suitable for use as an encoding system 100 or just a Huffman encoding module 114 of **Figure 1**.

[0069] The computer system 840 includes a processor 850, memory 855 and input/output capability 860 coupled to a system bus 65. The memory 855 is configured to store instructions which, when executed by the processor 850, perform the methods described herein. In addition, the memory 855 may be configured to store a set difference tables used by the Huffman encoding module 114. Input/output 860 also encompasses various types of computer-readable media, including any type of storage device that is accessible by the processor 850. One of skill in the art will immediately recognize that the term “computer-readable medium/media” further encompasses a carrier wave that encodes a data signal. It will also be appreciated that the system 840 is controlled by

operating system software executing in memory 855. Input/output and related media 860 store the computer-executable instructions for the operating system and methods of the present invention. The Huffman encoding module 114 shown in **Figure 1** may be a separate component coupled to the processor 850, or may be embodied in computer-executable instructions executed by the processor 850. In one embodiment, the computer system 840 may be part of, or coupled to, an ISP (Internet Service Provider) through input/output 860 to transmit or receive image data over the Internet. It is readily apparent that the present invention is not limited to Internet access and Internet web-based sites; directly coupled and private networks are also contemplated.

[0070] It will be appreciated that the computer system 840 is one example of many possible computer systems that have different architectures. A typical computer system will usually include at least a processor, memory, and a bus coupling the memory to the processor. One of skill in the art will immediately appreciate that the invention can be practiced with other computer system configurations, including multiprocessor systems, minicomputers, mainframe computers, and the like. The invention can also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network.

[0071] Various aspects of Huffman encoding have been described. Although specific embodiments have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that any arrangement which is calculated to achieve the same purpose may be substituted for the specific embodiments shown. This application is intended to cover any adaptations or variations of the present invention.